Array 11 (2021) 100074

Contents lists available at ScienceDirect

Array

journal homepage: www.elsevier.com/journals/array/2590-0056/open-access-journal

Wild Geese Algorithm: A novel algorithm for large scale optimization based on the natural life and death of wild geese

Mojtaba Ghasemi [a], Abolfazl Rahimnejad [b], Rasul Hemmati [c], Ebrahim Akbari [d], S. Andrew Gadsden [b,*]

[a] Department of Electronics and Electrical Engineering, Shiraz University of Technology, Shiraz, Iran
[b] Department of Engineering Systems and Computing, University of Guelph, Guelph, Canada
[c] Department of Electrical and Computer Engineering, Marquette University, Milwaukee, USA
[d] Department of Electrical Engineering, Faculty of Engineering, University of Isfahan, Isfahan, Iran

ARTICLE INFO

Keywords:
Large-scale global optimization (LSGO)
Wild Geese Algorithm (WGA)
Swarm-based method
Engineering optimization

ABSTRACT

In numerous real-life applications, nature-inspired population-based search algorithms have been applied to solve numerical optimization problems. This paper focuses on a simple and powerful swarm optimizer, named Wild Geese Algorithm (WGA), for large-scale global optimization whose efficiency and performance are verified using large-scale test functions of IEEE CEC 2008 and CEC 2010 special sessions with high dimensions $D = 100$, 500, 1000. WGA is inspired by wild geese in nature and models various aspects of their life such as evolution, regular cooperative migration, and fatality. The effectiveness of WGA for finding the global optimal solutions of high-dimensional optimization problems is compared with that of other methods reported in the previous literature. Experimental results show that the proposed WGA has an efficient performance in solving a range of large-scale optimization problems, making it highly competitive among other large-scale optimization algorithms despite its simpler structure and easier implementation. The source code of the proposed WGA algorithm is publicly available at github.com/ebrahimakbary/WGA.

1. Introduction

Many practical optimization problems, which are called Large Scale Global Optimization (LSGO) problems, deal with a lot of decision variables. Some practical LSGO problems are large-scale electronic systems design, scheduling problems, vehicle routing in large-scale traffic networks, and inverse problem chemical kinetics. Many real-world optimization problems involve optimization of a large number of control variables with various constraints. However, the classical mathematical programming methods do not generally provide good solutions for different optimization problems with different real-world complexities, due to the huge size of the problems [1]. The global optimization performance of the population-based algorithms often becomes weaker in such problems with increasing the dimension and complexity of the problem [2–4]. The practical large-scale optimization problems have been modeled with different benchmark test functions such as those presented in the CEC 2008 [5] and CEC 2010 [6].

Recently, many nature-inspired and population-based meta-heuristic optimization algorithms have been presented to deal with LSGO problems with different real-world complexities such as nonlinearity, non-smoothness, non-convexity, mixed-integer nature, non-differentiability, etc. Some new nature-inspired optimization algorithms for solving the practical large-scale optimization problems are listed in Table 1. It should be mentioned that, the boldface rows of this table, show the methods which were used in the comparative study with the proposed WGA.

Wild geese have a long-distance, coordinated and organized travel, which can be used as an inspiration for a very appropriate optimization algorithm for high-dimension problems. Based on the general model of wild geese' lives, a novel algorithm called Wild Geese Algorithm (WGA) is introduced in this paper, which have some main prominent characteristics compared to the previous algorithms including:

- It is simple with low computational burden, and its implementation is easily performed.
- It has proper and satisfactory power for different test functions, from different groups.

* Corresponding author.
E-mail address: gadsden@uoguelph.ca (S.A. Gadsden).

https://doi.org/10.1016/j.array.2021.100074
Received 23 December 2020; Received in revised form 1 May 2021; Accepted 16 June 2021
Available online 25 June 2021

Table 1
Summary of some new nature-inspired optimization algorithms for solving the practical large-scale optimization problems.

Ref.	Year	Abbreviation	Short Description	Dimensions under study	Real-world problem
[7]	**2008**	**MLCC**	**Multilevel Cooperative Coevolution**	**100, 500, 1000**	**No**
[8]	**2008**	**EPUS-PSO**	**Efficient Population Utilization Strategy for Particle Swarm Optimizer (PSO)**	**100, 500, 1000**	**No**
[9]	**2008**	**sep-CMA-ES**	**Covariance Matrix Adaptation Evolution Strategy having diagonal covariance matrix**	**100–1000**	**No**
[10]	2010	SOUPDE	Shuffle or update parallel differential evolution	50, 100, 200, 500, 100	No
[11]	2010	CCVIL	Cooperative Coevolution with Variable Interaction Learning	1000	No
[12]	**2010**	**•DECC-D •DECC-DML**	**•Differential Evolution with Cooperative Co-evolution using Delta-Grouping •Differential Evolution with Multilevel Cooperative Co-evolution using Delta-Grouping**	**100, 500, 1000**	**No**
[13]	2010	GOBL	Generalized Opposition-Based Learning	50, 100, 200, 500, 100	No
[14]	2011	TSVP	Tabu Search with Variable Partitioning	100, 400, 1000	No
[15]	2011	SP-UCI	Shuffled complex evolution with principal components analysis–University of California at Irvine	10, 50, 100, 1000	No
[16]	2012	LMDEa	Differential Evolution with Landscape Modality Detection and a Diversity Archive	1000	No
[17]	2012	DE-CCS	Differential Evolution Algorithm with Cooperative Coevolutionary Selection Operator	500,1000	No
[2]	**2012**	**CCPSO2**	**A new Cooperative Coevolving Particle Swarm Optimization with a new position update rule based on Cauchy and Gaussian distributions**	**1000**	**No**
[18]	2012	LSCBO	Large Scale Optimization Based on Co-ordinated Bacterial Dynamics and Opposite Numbers	100, 500, 1000	No
[19]	2013	GOjDE	A Generalized Opposition based Differential Evolution enhanced with a self-adapting parameter tuning strategy	100, 200, 500, 1000	No
[20]	2013	EOEA	A two-stage based ensemble optimization evolutionary algorithm	1000	Yes
[21]	2014	FT-DNPSO	PSO with dynamic neighborhood based on kernel fuzzy clustering and variable trust region methods	30, 100, 1000	No
[22]	**2014**	**CBCC1-DG CBCC2-DG DECC-DG**	**Two different versions of Contribution Based Cooperative Co-evolution and Differential Evolution with Cooperative Co-evolution, all with differential grouping**	**1000**	**No**
[23]	2015	CDE	Continuous Differential Evolution	200, 500, 1000	No
[24]	2015	CSO	A Competitive Swarm Optimizer	100, 500, 1000, 2000, 5000	No
[25]	2016	SOMAQI	Self Organizing Migrating Algorithm with Quadratic Interpolation	100, 500, 1000, 2000, 3000	No
[26]	2018	MWOA	A Modified Whale Optimization Algorithm	100, 300, 500, 1000	No
[27]	2019	EHO	Enhanced Elephant Herding Optimization with Novel Individual Updating Strategies	50, 100, 200, 500, 100	No
[28]	2019	SFO	Sailfish Optimizer	300	Yes
[29]	2019	PRO	Poor and rich optimization algorithm	300	Yes
[30]	2019	EBA	Ensemble Bat Algorithm	100, 500, 1000	No
[31]	**2019**	**EO**	**Equilibrium optimizer**	**10–200**	**Yes**
[32]	2020	NPO	Nomadic People Optimizer	100, 500, 2000	No
[33]	**2020**	**ISSA**	**An improved Social Spider Algorithm**	**100, 500, 1000**	**No**

It is worth mentioning that although the proposed WGA may seem similar to PSO, especially due to the existence of personal best and global best concepts, it has some thorough distinctions of structure and formulation, the main of which can be listed as follows:

1 In WGA, all solutions are sorted based on their objective values so that each member of population moves using information from its adjacent members in the sorted population.
2 In the proposed method, the formulation for calculating the velocity of each goose is completely different from the PSO and is based on the positions, velocities, and best positions of the goose and its adjacent geese in the sorted population, as well as the global best solution's position. While in PSO, the only parameter that is shared among all solutions is the position of the global best solution.
3 In the proposed WGA, two different solutions are generated per solution and are used for creating the next iteration's goose based on a mechanism similar to the crossover operator of differential evolution.
4 Finally, in the proposed algorithm, a population reduction policy is implemented which is accomplished by fatality (elimination) of the weakest goose of the population.

The rest of this paper is organized as follows. Section 2 presents the new proposed algorithm for large-scale optimization problems. Section 3 shows the experimental results. Finally, Section 4 presents the conclusions.

2. The proposed algorithm: *Wild Geese Algorithm*

In recent years, some new algorithm inspired from group movement and group search by animals have been proposed for large-scale global continuous optimization [1]. In this paper, based on the different phases of wild geese's lives, including their rhythmic and coordinated group migration, reproduction and evolution and also deaths in the population of geese, a new efficient algorithm, named as Wild Geese Algorithm (WGA), is presented for high-dimensional optimization problems. In Fig. 1, a group ordered migration based on the position of wild geese is shown. In general, the proposed WGA phases are as follows:

1 Ordered and coordinated group migration (or migration and displacement velocity phase)
2 Walking and searching for food by wild geese.
3 Reproduction and evolution of wild geese.
4 Death, migration and ordered evolution of wild geese.

First, an initial population of wild geese are created, so that the position vector of the i-th wild goose is equal to x_i. The best local position or personal best solution p_i and migration velocity $v_i$are determined. Then, all wild geese populations are sorted from the best to the worst according to their target function.

In this modeling strategy, each wild goose exploits information from its adjacent wild geese in the ordered population, and is directed by those individuals. The phases of WGA are further discussed in the subsequent subsections.

2.1. An ordered and coordinated group migration (or migration and displacement velocity phase)

As it is observed in Fig. 1, migration of wild geese is a group, coordinated, ordered and under control migration, which is based on reaching the upfront and adjacent individuals in the sorted population. Velocity and displacement equations according to the coordinated velocity of the geese are given in Eq. (1) and Eq. (2).

$$\begin{aligned} v_{i,d}^{Iter+1} &= \left(r_{1,d} \times v_{i,d}^{Iter} + r_{2,d} \times \left(v_{i+1,d}^{Iter} - v_{i-1,d}^{Iter}\right)\right) \\ &+ r_{3,d} \times \left(p_{i,d}^{Iter} - x_{i-1,d}^{Iter}\right) + r_{4,d} \times \left(p_{i+1,d}^{Iter} - x_{i,d}^{Iter}\right) \\ &+ r_{5,d} \times \left(p_{i+2,d}^{Iter} - x_{i+1,d}^{Iter}\right) - r_{6,d} \times \left(p_{i-1,d}^{Iter} - x_{i+2,d}^{Iter}\right) \end{aligned} \quad (1)$$

Fig. 1. An ordered and coordinated migration of wild geese.

where $x_{i,d}$, $p_{i,d}$, and $v_{i,d}$ are the dth dimension of the current position, the best position, and the current velocity of the ith wild goose, respectively. Note that in this study, $r_{k,d}, k = 1, 2, \ldots, 11$ are uniformly distributed random numbers between 0 and 1.

As observed in Eq. (1), the velocity and position changes of each wild goose (for instance i-th wild goose) depend on the velocities of their upfront and rear members, i.e ($v_{i+1}^{Iter} - v_{i-1}^{Iter}$) , and also to the positions of its adjacent members.

According to the model from the migration of wild geese in Fig. 2 and Eq. (1), the wild geese use information from their adjacent individuals in the sorted population, as patterns for their movement and navigation, and tend to reach those members (reduce their distances), i.e. $x_{i-1}^{Iter} \rightarrow p_i^{Iter}$, $x_i^{Iter} \rightarrow p_{i+1}^{Iter}$, $x_{i+1}^{Iter} \rightarrow p_{i+2}^{Iter}$, and $x_{i+2}^{Iter} \rightarrow - p_{i-1}^{Iter}$.

Additionally, the global best member is used as another guide for the movements of the whole flock; which is reflected in Eq. (2). This position change is carried out in an ordered form and coordinated with the upfront members in order to model the movement of all members as an ordered series, as shown in Figs. 1 and 2.

$$x_{i,d}^{v} = p_{i,d}^{Iter} + r_{7,d} \times r_{8,d} \times \left(\left(g_d^{Iter} + p_{i+1,d}^{Iter} - 2 \times p_{i,d}^{Iter}\right) + v_{i,d}^{Iter+1}\right) \quad (2)$$

where g_d is the global best position among all members.

2.2. Walking and searching for food by wild geese

This step is modeled in such a way that the i-th wild goose moves towards its upfront member, i.e. the (i+1)-th goose ($p_i^{Iter} \rightarrow p_{i+1}^{Iter}$). In another word, the i-th goose tries to reach the (i+1)-th goose ($p_{i+1}^{Iter} - p_i^{Iter}$). The equation for walking and searching for food by the wild goose, x_i^W is as follows:

$$x_{i,d}^{w} = p_{i,d}^{Iter} + r_{9,d} \times r_{10,d} \times \left(p_{i+1,d}^{Iter} - p_{i,d}^{Iter}\right) \quad (3)$$

2.3. Reproduction and evolution of wild geese

Another stage of wild geese's life is reproduction and evolution. In this paper, its modeling is performed so that a combination between migration equation (x_i^V) and walking and search for food equation (x_i^W) is used. The Cr value for the proposed WGA algorithm is 0.5 in total simulations.

$$x_{i,d}^{Iter+1} = \begin{cases} x_{i,d}^{v} \text{ if } & r_{11,d} \le Cr \\ x_{i,d}^{w} & \text{otherwise.} \end{cases} \quad (4)$$

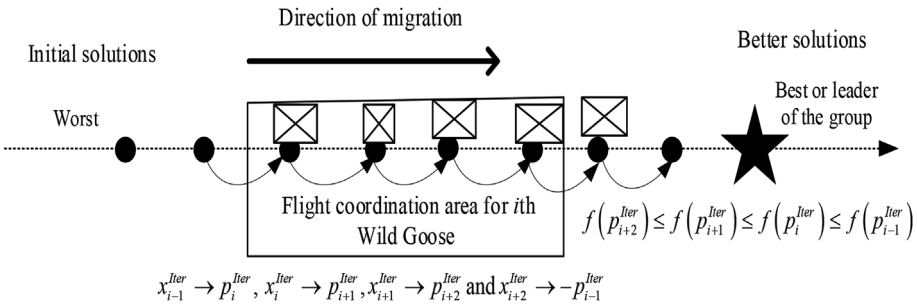

Fig. 2. The model of ordered and coordinated group migration of wild geese.

2.4. Death, migration and ordered evolution

The previous experiments from the literature show that for different optimization algorithms the population number and the iteration number do not have the same level of influence on solving every types of problems. For some functions, the size of algorithm's population is more important and more effective than the number of algorithm's iterations (e.g. F2 and F3 functions), and for some other functions the number of algorithm's iterations is more important and more effective than the size of WGA algorithm's population (e.g. F7 and F8 functions). In this paper, to overcome this problem and establish a compromised solution, the death phase is employed in order to balance algorithm performance for all test functions. In this phase, the algorithm starts with the maximum population number $Np^{initial}$ and during the algorithm iterations, the weaker members will be removed from the population based on Eq. (5) and the population size will decrease linearly so that it reaches its final value Np^{final} in the final iteration.

$$Np = \text{round}\left(Np^{initial} - \left(\left(Np^{initial} - Np^{final}\right) * \left(\frac{FEs}{FEs_{\max}}\right)\right)\right) \tag{5}$$

where FEs and FEs_{max} are the number of function evaluations and its maximum.

Algorithm 1
Demonstrates the optimization process of WGA.

Algorithm 1:
to set values of the control parameters of WGA;
to generate the initial population (whose number are equal to $Np^{initial}$) and $V_i^{Iter=1} = [0]$;
to evaluate the fitness of each population individual and $FEs = Np^{initial}$;
to find the personal best position of all particles $Np^{initial}$ ($i = 1, 2, \ldots, Np^{initial}$) in swarm $P_i$and the global best position G;
while the FEs till FEs_{max} **do**
  Wild Goose populations are arranged from the best to the worst according to Fig. 2;
  for $i = 1$ (best) to Np (worst) **do**
    Select the sorted members $i-1$th, $i+1$th, and $i+2$th;
    {** An ordered and coordinated group migration based on Eq. (1) and Eq. (2) **}
    for $d = 1$ to D **do**
      $V_i^{Iter+1} \leftarrow$ Eq. (1);
    end for
    for $d = 1$ to D **do**
      $x_{i,d}^{V} \leftarrow$ Eq. (2);
    end for
    {** Walking and search geese Eq. (3) **}
    for $d = 1$ to D **do**
      $x_{i,d}^{W} \leftarrow$ Eq. (3);
    end for
    {** Reproduction and evolution Eq. (4) **}
    for $d = 1$ to D **do**

(*continued on next column*)

Algorithm 1 (*continued*)

      $X_i^{Iter+1} \leftarrow$ Eq. (4);
    end for
    if $x_{i,d}^{Iter+1} < x_d^{min}$
      $x_{i,d}^{Iter+1} \leftarrow x_d^{min}$;
    end if
    if $x_{i,d}^{Iter+1} > x_d^{max}$
      $x_{i,d}^{Iter+1} \leftarrow x_d^{max}$;
    end if
    to evaluate the fitness of X_i^{Iter+1}
    if $f(X_i^{Iter+1}) \le f(P_i^{Iter})$
      $P_i^{Iter+1} \leftarrow X_i^{Iter+1}$;
    end if
    if $f(P_i^{Iter+1}) \le f(G)$
      $G \leftarrow P_i^{Iter+1}$;
    end if
  end for
  $FEs = FEs + Np$;
  $Np \leftarrow$ Eq. (5);
end while

3. Results and analysis of experimental evaluation studies

In this section, 20 widely used large scale test functions are exploited to show the efficiency and performance of the proposed algorithm. The formulation and characteristics of all CEC 2010 benchmark test functions are listed in Ref. [6].

The performance and robustness of WGA for solving real and large-scale optimization problems are characterized by two indices: 1) the mean of best values of test function (Mean), and 2) the standard deviation (Std) indices.

Test functions include 1. Separable functions ($F1 - F3$), 2. Single-group m-nonseparable functions ($F4 - F8$), 3. $\frac{D}{2m}$-group m-nonseparable functions ($F9 - F13$), 4. $\frac{D}{m}$-group m-nonseparable functions ($F14 - F18$), and 5. non-separable functions ($F19 - F20$), where m is the number of variables in each non-separable subcomponent, and D and m are assumed as 1000 and 50, respectively. To show the efficiency of WGA, in all simulations of this paper, 25 independent simulations are used in each section for every test function, as in Refs. [6,22]. Furthermore, in all simulations, the maximum number of fitness evaluations FEs_{max} is 3×10^6. In all tables, the + sign means the algorithm outperforms WGA, the – sign means WGA outperforms the algorithm, and the = sign means WGA and the considered algorithm yield the same solution for the given problem. It should be mentioned that, in all results tables, the boldface is used to emphasize the algorithm that achieves the best Mean index value for each problem.

3.1. Experimental setup

3.1.1. Influence of death phase on WGA performance

At first, to show the performance of the population reduction by death

Table 2
Average fitness values and standard deviations of results for test functions over 25 independent runs.

F	WGA, $Np = 30$	WGA, $Np = 120$	WGA
*F*1	1.68E-21	2.33E-24	**1.05E-26**
	7.71E-22	1.58E-24	**2.56E-26**
	3	2	**1**
*F*2	7.78E+03	**2.18E + 03**	2.28E+03
	7.95E+01	**1.14E + 01**	4.58E+01
	3	**1**	2
*F*3	1.00E+01	**1.17E-13**	1.47E-13
	1.25E+01	**7.40E-15**	8.94E-15
	3	**1**	2
*F*4	**3.81E + 11**	9.99E+11	5.15E+11
	1.63E + 11	1.05E+11	7.89E+10
	1	3	2
*F*5	9.55E+07	5.74E+07	**5.47E + 07**
	7.04E+06	3.68E+06	**7.93E + 06**
	3	2	**1**
*F*6	1.98E+01	3.56E-09	**3.55E-09**
	2.50E-02	1.40E-15	**5.48E-14**
	3	2	**1**
*F*7	**8.01E-02**	4.47E+03	4.60E+00
	2.00E-02	1.69E+03	6.28E+00
	1	3	2
*F*8	**8.60E + 06**	4.30E+07	9.16E+06
	3.16E + 05	2.74E+07	8.79E+06
	1	3	2
*F*9	2.54E+07	4.55E+07	**2.21E + 07**
	1.33E+06	5.50E+06	**1.51E + 06**
	2	3	**1**
*F*10	4.67E+03	**1.76E + 03**	2.64E+03
	1.60E+02	**2.48E + 01**	2.70E+01
	3	**1**	2
*F*11	8.94E+01	**2.34E-13**	3.06E-13
	7.77E+00	**1.07E-14**	5.48E-14
	3	**1**	2
*F*12	**1.62E + 03**	3.25E+04	4.15E+03
	1.30E + 02	1.40E+03	2.40E+02
	1	3	2
*F*13	9.11E+02	9.87E+02	**6.87E + 02**
	1.93E+02	1.50E+02	**2.63E + 01**
	2	3	**1**
*F*14	**7.51E + 07**	1.52E+08	7.67E+07
	5.36E + 06	1.24E+07	4.55E+06
	1	3	2
*F*15	5.28E+03	4.21E+03	**3.14E + 03**
	3.79E+02	1.01E+02	**5.42E + 01**
	3	2	**1**
*F*16	2.69E+02	7.63E+00	**3.79E + 00**
	1.37E+01	2.95E+00	**6.26E-01**
	3	2	**1**
*F*17	**1.41E + 04**	1.47E+05	3.74E+04
	6.23E + 02	7.77E+03	1.36E+02
	1	3	2
*F*18	2.11E+03	4.15E+03	**1.52E + 03**
	1.47E+03	1.56E+03	**2.93E + 02**
	2	3	**1**
*F*19	**8.73E + 05**	1.35E+06	1.04E+06
	1.03E + 05	5.17E+04	2.85E+04
	1	3	2
*F*20	1.58E+03	1.15E+03	**1.04E + 03**
	7.71E+01	2.42E+01	**8.18E + 01**
	3	2	**1**
Nb/Nw/Mr	**7**/10/2.15	4/10/2.3	**10/0/1.55**

of Wild Geese, WGA is tested without considering the death phase and is tested with a large population $Np = 120$ and a small population $Np = 30$. The suitable results were compared with those of WGA (considering population reduction from $Np = 120$ ($Np^{initial}$=120) to $Np = 30$ (Np^{final}=30) using Eq. (5), where the results obtained for each function are listed in Table 2. The results demonstrate that the proposed death phase improves the efficiency of WGA for high-dimensional problems. The positive influence of death phase can be especially observed for test functions *F*3, *F*6, *F*7, *F*11, *F*12, *F*16, and *F*17. Moreover, the convergence characteristics of this algorithm for 6 different functions of various types are depicted in Fig. 3, which verify the effectiveness of implementing death phase in WGA.

3.1.2. Why $Cr = 0.5$ in WGA for all test functions?

In this paper $Cr = 0.5$ is used for all simulations. To select a suitable value for Cr four different constant values other than 0.5, i.e. 0.1, 0.25, 0.75 and 0.9 are tested, whose results are presented in Table 3. As observed, the constant value 0.5 is the best value for different test functions of CEC 2010. It should be mentioned that in all simulation results tables, three values are reported for optimizing each test function with each algorithm; the first two demonstrate the average and standard deviation of fitness values of the obtained results. The third value shows the rank of that algorithm in terms of the mean index. Furthermore, three parameters are reported for each algorithm in all tables, i.e. N_b, N_w, M_r. N_b and N_w are the number of times the algorithm yields the best and the worst mean index, respectively; and M_r is the average rank of the algorithm achieved in solving all considered test functions.

3.2. Comparing WGA with recent optimization algorithms

3.2.1. CEC 2008 test functions

In this section, the results of WGA are compared with those of a series of the recently proposed optimization algorithms for large-scale problems from CEC 2008 test functions with different high dimensions including $D = 100$, $D = 500$ and $D = 1000$. The formulation and characteristics of CEC′08 benchmark test functions are listed in Ref. [5] and Table 4:

Two indices are exploited in this study to characterize the performance and robustness of WGA for solving real and large-scale optimization problems with different dimensions: 1) mean of the best values of test function (Mean), and 2) standard deviation (Std). Tables 5–7 show the final best solutions of test functions' optimization by WGA and those of large scale optimization algorithms including CSO [24], CCPSO2 [2], sep-CMA-ES [9], MLCC [7], and EPUS-PSO [8]. As seen, the proposed WGA is able to provide very efficient and competitive results in solving real and large-scale problems compared with the previously proposed algorithms. WGA proves itself as a promising technique for real and large scale shifted unimodal and multimodal optimization problems.

3.2.2. CEC 2010 test functions

As mentioned in the introduction section of this paper, numerous researches have been recently performed to achieve some algorithms and meta-heuristic optimization methods for high-dimension optimization problems. These studies and many other methods have been introduced to find a simple and quick method with the low computational burden. In Table 8, the results of previous researches for 20 different test functions of CEC 2010 with $D = 1000$ are summarized [22], which was obtained with the same conditions as those of WGA. The summarized algorithms in Table 8 include MLCC algorithm [7], differential evolution with cooperative co-evolution and delta grouping DECC-D and DECC-DML [12], contribution based cooperative co-evolution and differential grouping CBCC1-DG and CBCC2-DG [22], differential evolution with cooperative co-evolution and random grouping (DECC-DG) [22]. The last two rows of Table 8 present the comparative indices for these algorithms.

The WGA algorithm has achieved the best results in 12 of 20 functions, i.e. *F*4, *F*5, *F*6, *F*7, *F*9, *F*10, *F*13, *F*14, *F*15, *F*17, *F*18, and *F*19. In addition, for none of the test functions WGA has the worst results. Moreover, WGA reaches the best average rank (M_r). The proposed algorithm (WGA) outperforms MLCC algorithm in 18 out of 20 functions; only for the first two functions MLCC algorithm performs better. For the first function the average value of WGA is very close to that of MLCC algorithm. MLCC algorithm has different results for different test functions and has the worst results for 6 out of 20 functions. However, the proposed algorithm has acceptable and suitable results for most of the

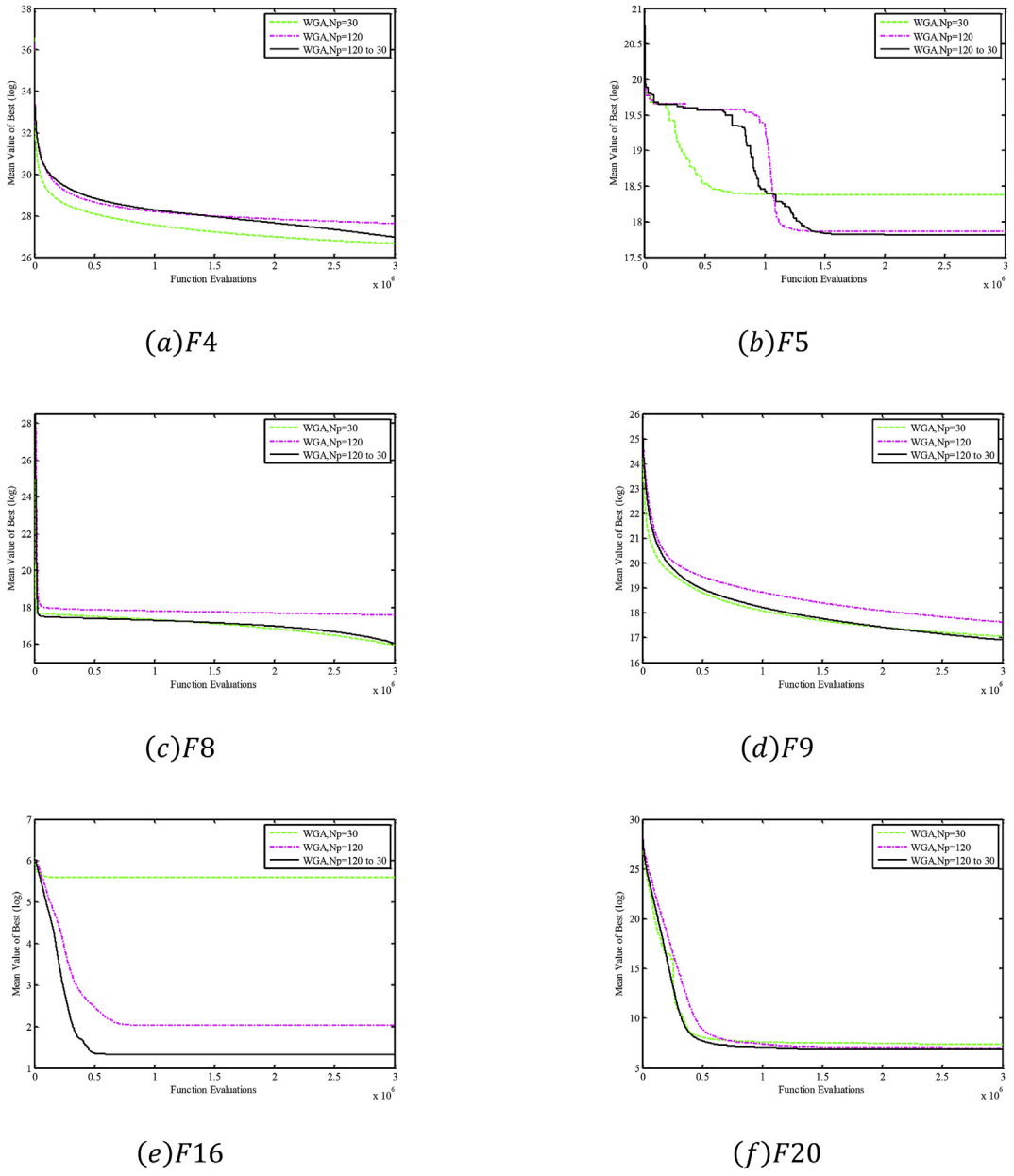

Fig. 3. Average convergence of WGA on nine selected test functions over 25 independent runs.

test functions and dispersion of its results are less than those of the other algorithms. The comparison between WGA and DECC-D algorithm shows that WGA performs better for 18 out of 20 functions. Nonetheless, for functions $F2$ and $F20$, it gives a worse result compared to that of DECC-D. For function $F2$, the average value of WGA is very close to that obtained from DECC-D algorithm. Furthermore, DECC-D algorithm does not provide a good quality solution for different test functions, for example for $F2$ and $F20$ it has suitable results, but for $F5-F8$, $F10-F12$, and $F15-F17$ its results are not acceptable compared to those of other algorithms. Although DECC-DML algorithm outperforms WGA for five test functions, it has the worst solution for six functions. CBCC1-DG and CBCC2-DG algorithms are more successful than WGA for two and three functions, respectively; however, CBCC1-DG gives the best result for none of the functions and CBCC2-DG yields the best result for only function. DECC-DG algorithm performs better than WGA for 2 out of 20 test functions; however, it gives the worst solution for 4 test functions among all algorithms.

3.2.3. *Test on real-world optimization problems*

Here, the effectiveness of the proposed algorithm (WGA) was investigated compared to genetic algorithm (GL-25) [34], DE with strategy adaptation (SaDE) [35], DE with control components and composite trial vector generation approaches (CoDE) [36], Standard particle swarm optimization (SPSO2013) [37], and heterogeneous comprehensive learning PSO with improved exploitation and exploration (HCLPSO) [38] on real-world usages including estimating the factor for frequency-modulated (FM) sound waves [39] and large-scale reliability-redundancy allocation optimization (RRAO) of a gas turbine [40].

1) Estimating the factor for frequency modulated sound waves

The greatly complex multimodal frequency-modulated (FM) sound synthesis optimizing problem plays a key role in various modern music systems for estimating the optimal factors of a FM sound wave synthesis [39]. The estimation of optimal factors of an FM sound wave synthesis is

Table 3
Average fitness values and standard deviations on test functions over 25 independent runs.

F	*Cr* = 0.1	*Cr* = 0.25	*Cr* = 0.75	*Cr* = 0.9	*Cr* = 0.5
F1	3.12E+07	3.77E-06	5.24E+09	5.00E+10	**1.05E-26**
	7.41E+07	1.63E-07	1.03E+09	2.29E+10	**2.56E-26**
	3,–	2,–	4,–	5,–	1
F4	1.17E+12	7.26E+11	4.96E+13	2.48E+14	**5.15E+11**
	7.40E+11	9.64E+10	8.22E+13	5.31E+13	**7.89E+10**
	3,–	2,–	4,–	5,–	1
F9	1.39E+10	9.04E+07	1.03E+10	6.71E+10	**2.21E+07**
	8.22E+09	2.76E+08	7.50E+09	2.56E+10	**1.51E+06**
	4,–	2,–	3,–	5,–	1
F14	3.04E+10	2.71E+09	3.19E+09	1.55E+11	**7.67E+07**
	2.61E+10	1.26E+09	4.54E+09	1.23E+11	**4.55E+06**
	4,–	2,–	3,–	5,–	1
F20	1.50E+10	**1.03E+03**	1.17E+11	6.53E+11	1.04E+03
	4.20E+09	**5.15E+01**	3.73E+09	3.90E+09	8.18E+01
	3,–	1,+	4,–	5,–	2
−/+/ =	5/0/0	4/1/0	5/0/0	5/0/0	–
Nb/Nw/Mr	0/0/3.4	1/0/1.8	0/0/3.6	0/0/5	4/0/1.2

Table 4
Summary of CEC 08 Special Session benchmark test functions [5] for large scale global optimization.

Function	Name	Properties	Search space	Global optimum
f_1	Shifted Sphere	Unimodal, separable, scalable	[-100, 100]	0
f_2	Shifted Schwefel's	Unimodal, non-separable, scalable	[-100, 100]	0
f_3	Shifted Rosenbrock's	Multimodal, non-separable, scalable	[-100, 100]	0
f_4	Shifted Rastrigin's	Multimodal, separable, scalable	[-5, 5]	0
f_5	Shifted Griewank's	Multimodal, non-separable, scalable	[-600, 600]	0
f_6	Shifted Ackley's	Multimodal, separable, scalable	[-32, 32]	0

an optimization problem with D decision variables. In this work, the case of $D = 6$ is only considered in accordance with [41,42]. Six components are included in the 6-dimensional parameter vector as $x = [x_1(a_1), x_2(\omega_1), x_3(a_2), x_4(\omega_2), x_5(a_3), x_6(\omega_3)]$ ranging between 6.35 and 6.5 for all variables. The equations provided for the target and approximated sound waves for t defined in range of 1–100 are as follows [42]:

$$y(t) = x_1 \sin(x_2 t\theta + x_3 \sin(x_4 t\theta + x_5 \sin(x_6 t\theta))), \tag{6}$$

$$y_0(t) = 1.0*\sin(0.5t\theta - 1.5 * \sin(4.8t\theta + 2.0 * \sin(4.9t\theta))), \tag{7}$$

where $\theta = \frac{2\pi}{100}$

The optimization problem objective function is considered as the sum of squared errors between $y(t)$(the approximated wave) and $y_0(t)$ (the target wave) with optimal value $f(x) = 0$ as follows:

$$f(x) = \sum_{t=0}^{100} (y(t) - y_0(t))^2. \tag{8}$$

2) RRAO constrained problem:

The nonlinear reliability-redundancy constrained optimization problems are mainly aimed at enhancing the system reliability (maximizing the overall system reliability) through optimizing element reliabilities vector ($r = (r_1, r_2, \ldots, r_m)$) and redundancy assignment numbers vector ($n = (n_1, n_2, \ldots, n_m)$) for subsystems of the system. It is possible to formulate this problem as a nonlinear mixed-integer programming model by choosing the system reliability as the objective function to be maximized subjecting to several nonlinear constraints as follows [40]:

$$\text{Maximize } R_s = f(r,\ n), \tag{9}$$

$$\begin{aligned} &\text{subject to } g(r,\ n) \le l, \\ &0 \le r_d \le 1,\ n_d \in Z^+,\ 0 \le d \le m. \end{aligned} \tag{10}$$

where Z^+ is the set of positive integers, R_s represents the reliability of various systems, *f(.)* and *g(.)* denote for the objective and constraint functions of RRAO problem for the total parallel-series systems, respectively, from which *g(.)* is usually related to the system cost, weight and volume. $n = (n_1, n_2, \ldots, n_m)$ and $r = (r_1, r_2, \ldots, r_m)$ show the redundancy allocation numbers and component reliabilities vectors for system's subsystems including m subsystems, respectively. Moreover, l shows the limit of the system resources.

The overspeed detection was continually offered by the mechanical and electrical systems. By occurring an overspeed, the fuel source must be stopped through control valves (V_1 to V_m). Fig. 4 represents a gas turbine's overspeed protection system for RRAO optimizing the mixed-integer non-linear problem. The large-scale test structure involves 40

Table 5
Results obtained by optimization algorithms for dimension 100 over 25 independent runs.

F	*D* = 100							
	CCPSO2 [2]	CSO [24]	sep-CMA-ES [9]	MLCC [7]	EPUS-PSO [8]	ISSA [33]	EO [31]	WGA
F1	7.73E-14	9.11E-29	9.02E-15	6.82E-14	7.47E-01	**0**	1.31E-20	**0**
	3.23E-14	1.10E-28	5.53E-15	2.32E-14	1.70E-01	**0**	5.01E-20	**0**
	6,-	2,-	4,-	5,-	7,-	**1, =**	3.-	**1**
F2	6.08E+00	3.35E+01	2.31E+01	2.53E+01	1.86E+01	8.31E+01	4.29E+01	**2.14E-05**
	7.83E+00	5.38E+00	1.39E+01	8.73E+00	2.26E+0	1.91 E+01	3.69E+00	**3.08E-05**
	2,-	6,-	4,-	5,-	3,-	8,-	7,-	**1**
F3	4.23E+02	3.90E+02	**4.31E + 00**	1.50E+02	4.99E+03	1.68E+02	9.21E+01	1.04E+02
	8.65E+02	5.53E+02	**1.26E + 01**	5.72E+01	5.35E+03	9.46E+01	8.97E+01	4.01E+01
	7,-	6,-	**1, +**	4,-	8,-	5,-	2,+	3
F4	3.98E-02	5.60E+01	2.78E+02	**4.39E-13**	4.71E+02	5.00E+00	6.04E+02	1.25E+02
	1.99E-01	7.48E+00	3.43E+01	**9.21E-14**	5.94E+01	6.60E+00	8.52E+01	1.41E+01
	2,+	4,+	6,-	**1,+**	7,-	3,+	8,-	5
F5	3.45E-03	**0**	2.96E-04	3.41E-14	3.72E-01	**0**	9.58E-02	**0**
	4.88E-03	**0**	1.48E-03	1.16E-14	5.60E-02	**0**	1.02E-01	**0**
	4,-	**1, =**	3,-	2,-	6,-	**1, =**	5,-	**1**
F6	1.44E-13	**1.20E-014**	2.12E+01	1.11E-13	2.06E+00	2.09E+01	2.05E+01	1.39E-014
	3.06E-14	**1.52E-015**	4.02E-01	7.87E-15	4.40E-01	2.99E-02	1.73E-01	1.23E-015
	4,-	**1,+**	8,-	3,-	6,-	7,-	5,-	2
−/+/ =	5/1/0	3/2/1	5/1/0	5/1/0	6/0/0	3/1/2	5/1/0	–
Nb/Nw/Mr	0/0/4.167	2/0/3.333	1/1/4.333	1/0/3.333	0/4/6.167	**2/1/4.167**	0/1/5	**3/0/2.333**

Table 6
Results obtained by optimization algorithms for dimension 500 over 25 independent runs.

F	$D = 500$							
	CCPSO2 [2]	CSO [24]	sep-CMA-ES [9]	MLCC [7]	EPUS-PSO [8]	ISSA [33]	EO [31]	WGA
*F*1	3.00E-13	6.57E-23	2.25E-14	4.30E-13	8.45E+01	9.90E-28	4.14E-04	**0.00E + 00**
	7.96E-14	3.90E-24	6.10E-15	3.31E-14	6.40E+00	9.95E-28	3.87E-04	**0.00E + 00**
	5,-	3,-	4,-	6,-	8,-	2,-	7,-	**1**
*F*2	5.79E+01	**2.60E + 01**	2.12E+02	6.67E+01	4.35E+01	2.66E+02	9.34E+01	5.73E+01
	4.21E+01	**2.40E + 00**	1.74E+01	5.70E+00	5.51E-01	1.92E+01	3.01E-01	8.72E+00
	4,-	**1,**+	7,-	5,-	2,+	8,-	6,-	3
*F*3	7.24E+02	5.74E+02	**2.93E + 02**	9.25E+02	5.77E+04	8.31E+14	1.95E+03	5.22E+02
	1.54E+02	1.67E+02	**3.59E + 01**	1.73E+02	8.04E+03	3.11E+14	1.04E+03	3.60E+01
	6,-	4,-	**1,**+	7,-	5,-	8,-	3,-	**2**
*F*4	3.98E-02	3.19E+02	2.18E+03	**1.79E-11**	3.49E+03	2.07E+03	3.78E+03	1.25E+02
	1.99E-01	2.16E+01	1.51E+02	**6.31E-11**	1.12E+02	5.38E+02	1.46E+02	1.41E+01
	2,+	4,-	6,-	**1,**+	7,-	5,-	8,-	3
*F*5	1.18E-03	**2.22E-16**	7.88E-04	2.13E-13	1.64E+00	4.48E-02	2.42E-01	4.12E-16
	4.61E-03	**0.00E + 00**	2.82E-03	2.48E-14	4.69E-02	1.29E-01	6.11E-01	5.36E-17
	5,-	**1,**+	4,-	3,-	8,-	6,-	7,-	2
*F*6	5.34E-13	4.13E-13	2.15E+01	5.34E-13	6.64E+00	2.14E+01	2.06E+01	**5.77E-14**
	8.61E-14	1.10E-14	3.10E-01	7.01E-14	4.49E-01	1.70E-02	3.35E-01	**1.58E-15**
	3,-	2,-	7,-	3,-	4,-	6,-	5,-	**1,**+
−/+/ =	5/1/0	4/2/0	5/1/0	5/1/0	5/1/0	6/0/0	6/0/0	–
Nb/Nw/Mr	0/0/4.167	2/0/2.5	1/2/4.833	1/1/4.167	0/3/5.667	0/2/5.833	0/1/6	**2/0/2**

Table 7
Results obtained by optimization algorithms for dimension $D = 1000$ over 25 independent runs.

F	$D = 1000$							
	CCPSO2 [2]	CSO [24]	sep-CMA-ES [9]	MLCC [7]	EPUS-PSO [8]	ISSA [33]	EO [31]	WGA
*F*1	5.18E-13	1.09E-21	7.81E-15	8.46E-13	5.53E+02	2.09E-18	1.35E+04	**1.75E-28**
	9.61E-14	4.20E-23	1.52E-15	5.01E-14	2.86E+01	3.95E-18	6.94E+03	**1.27E-28**
	5,-	2,-	4,-	6,-	7,-	3,-	8,-	**1**
*F*2	7.82E+01	**4.15E + 01**	3.65E+02	1.09E+02	4.66E+01	3.10E+02	1.64E+02	7.43E+01
	4.25E+01	**9.74E-01**	9.02E+00	4.75E+00	4.00E-01	1.38E+01	1.16E+02	4.89E+00
	4,-	**1,**+	8,-	5,-	2,+	7,-	6,-	3
*F*3	1.33E+03	1.01E+03	**9.10E + 02**	1.80E+03	8.37E+05	2.17E+15	2.58E+09	1.00E+03
	2.63E+02	3.02E+01	**4.54E + 01**	1.58E+02	1.52E+05	6.89E+13	2.63E+09	8.25E+01
	4,-	3,-	**1,**+	5,-	6,-	8,-	7,-	2
*F*4	1.99E-01	6.89E+02	5.31E+03	**1.37E-10**	7.58E+03	1.49E+04	7.79E+03	2.52E+03
	4.06E-01	3.10E+01	2.48E+02	**3.37E-10**	1.51E+02	1.93E+03	1.01E+02	1.34E+02
	2,+	3,+	5,-	**1,**+	6,-	8,-	7,-	4
*F*5	1.18E-03	**2.26E-16**	3.94E-04	4.18E-13	5.89E+00	3.10E-01	4.07E+01	1.22E-15
	3.27E-03	**2.18E-17**	1.97E-03	2.78E-14	3.91E-01	4.51E-01	5.39E+01	2.91E-16
	5,-	**1,**+	4,-	3,-	7,-	6,-	8,-	2
*F*6	1.02E-12	1.21E-12	2.15E+01	1.06E-12	1.89E+01	2.15E+01	2.05E+01	**1.21E-13**
	1.68E-13	2.64E-14	3.19E-01	7.68E-14	2.49E+00	7.70E-03	1.40E-01	**5.18E-15**
	2,-	4,-	5,-	3,-	6,-	8,-	7,-	**1**
−/+/ =	5/1/0	3/3/0	5/1/0	5/1/0	5/1/0	6/0/0	6/0/0	–
Nb/Nw/Mr	0/0/3.667	2/0/2.33	1/1/4.5	1/0/3.833	0/0/5.667	0/3/6.667	0/2/7.167	2/0/2.167

decision variables (m*2 = 40). The input factors and data for the large-scale test system are provided in Ref. [43] with 20 subsystems.

It is possible to formulate this reliability optimization problem as:

$$\begin{aligned} &\text{Maximize } f_5(r,\ n) = \prod_{d=1}^{m}\left[1-(1-r_d)^{n_d}\right]. \\ &0.5 \le r_d \le \left(1-10^{-6}\right),\ 0 \le d \le m \\ &1 \le n_d \le 10, \in Z^+. \end{aligned} \tag{11}$$

The system constraints include:

1) The combined weight, volume, and redundancy allocation constraint$g_1(r,\ n)$:

$$g_1(r,\ n) = \sum_{d=1}^{m} v_d^2 n_d^2 \le V \tag{12}$$

where v_d shows the volume of dth subsystem for all components and V represents the upper volume limit of the products of the subsystem.

2) The system cost limitation$g_2(r,\ n)$:

$$\begin{aligned} &g_2(r,\ n) = \sum_{d=1}^{m} C(r_d)\left[n_d + e^{0.25 n_d}\right] \le C, \\ &C(r_d) = \alpha_d\left(-\frac{T}{\ln r_d}\right)^{\beta_d}. \end{aligned} \tag{13}$$

where, C shows the upper cost limit of the system, $C(r_d)$ is the cost for all element with reliability r_d at dth stage, and T is the operating time in which the components are working.

3) The system weight limitation$g_3(r,\ n)$:

$$g_3(r,\ n) = \sum_{d=1}^{m} w_d n_d e^{0.25 n_d} \le W \tag{14}$$

Table 8
Average fitness values and standard deviations on CEC 2010 functions over 25 independent runs.

F	MLCC [7]	DECC-D [12]	DECC-DML [12]	CBCC1-DG [22]	CBCC2-DG [22]	DECC-DG [22]	WGA
*F*1	**1.53E-27**	1.01E-24	1.93E-25	1.32E+04	8.34E+03	5.47E+03	1.05E-26
	7.66E-27	1.40E-25	1.86E-25	6.25E+04	3.41E+04	2.02E+04	2.56E-26
	1,+	4,-	3,-	7,-	6,-	5,-	2
*F*2	**5.57E-01**	2.99E+02	2.17E+02	4.44E+03	4.44E+03	4.39E+03	2.28E+03
	2.21E + 00	1.92E+01	2.98E+01	1.60E+02	1.80E+02	1.97E+02	4.58E+01
	1,+	3,+	2,+	6,-	6,-	5,-	4
*F*3	9.88E-13	1.81E-13	**1.18E-13**	1.66E+01	1.67E+01	1.67E+01	1.47E-13
	3.70E-12	6.68E-15	**8.22E-15**	3.79E-01	3.28E-01	3.34E-01	8.94E-15
	4,-	3,-	**1**,+	5,-	6,-	6,-	2
*F*4	9.61E+12	3.99E+12	3.58E+12	2.31E+12	2.36E+12	4.79E+12	**5.15E + 11**
	3.43E+12	1.30E+12	1.54E+12	7.43E+11	7.92E+11	1.44E+12	**7.89E + 10**
	7,-	5,-	4,-	2,-	3,-	6,-	**1**
*F*5	3.84E+08	4.16E+08	2.98E+08	1.35E+08	1.36E+08	1.55E+08	**5.47E + 07**
	6.93E+07	1.01E+08	9.31E+07	2.18E+07	2.46E+07	2.17E+07	**7.93E + 06**
	6,-	7,-	5,-	2,-	3,-	4,-	**1**
*F*6	1.62E+07	1.36E+07	7.93E+05	1.65E+01	1.64E+01	1.64E+01	**3.55E-09**
	4.97E+06	9.20E+06	3.97E+06	3.99E-01	3.46E-01	2.71E-01	**5.48E-14**
	6,-	5,-	4,-	3,-	2,-	2,-	**1**
*F*7	6.89E+05	6.58E+07	1.39E+08	1.81E+04	1.35E+04	1.16E+04	**4.60E + 00**
	7.37E+05	4.06E+07	7.72E+07	4.59E+04	3.92E+04	7.41E+03	**6.28E + 00**
	5,-	6,-	7,-	4,-	3,-	2,-	**1**
*F*8	4.38E+07	5.39E+07	3.46E+07	3.34E+06	**8.70E + 05**	3.04E+07	9.16E+06
	3.45E+07	2.93E+07	3.56E+07	2.29E+06	**1.71E + 06**	2.11E+07	8.79E+06
	7,-	6,-	5.-	2,+	**1**,+	4,-	3
*F*9	1.23E+08	6.19E+07	5.92E+07	6.79E+07	7.97E+07	5.96E+07	**2.21E + 07**
	1.33E+07	6.43E+06	4.71E+06	6.92E+06	1.08E+07	8.18E+06	**1.51E + 06**
	7,-	4,-	2,-	5,-	6,-	3,-	**1**
*F*10	3.43E+03	1.16E+04	1.25E+04	4.01E+03	4.04E+03	4.52E+03	**2.64E + 03**
	8.72E+02	2.68E+03	2.66E+02	1.37E+02	1.21E+02	1.41E+02	**2.70E + 01**
	2,-	6,-	7,-	3,-	4,-	5,-	**1**
*F*11	1.98E+02	4.76E+01	**1.80E-13**	1.05E+01	1.03E+01	1.03E+01	**3.06E-13**
	6.98E-01	9.53E+01	**9.88E-15**	9.31E-01	8.47E-01	1.01E+00	**5.48E-14**
	6,-	5,-	**1**,+	4,-	3,-	3,-	**2**
*F*12	3.49E+04	1.53E+05	3.79E+06	4.19E+03	4.00E+03	**2.52E + 03**	4.15E+03
	4.92E+03	1.23E+04	1.50E+05	1.25E+03	8.63E+02	**4.86E + 02**	2.40E+02
	5,-	6,-	7,-	4,-	2,+	**1**,+	3
*F*13	2.08E+03	9.87E+02	1.14E+03	9.10E+03	4.54E+03	4.54E+06	**6.87E + 02**
	7.27E+02	2.41E+02	4.31E+02	3.75E+03	1.91E+03	2.13E+06	**2.63E + 01**
	4,-	2,-	3,-	6,-	5,-	7,-	**1**
*F*14	3.16E+08	1.98E+08	1.89E+08	3.64E+08	3.69E+08	3.41E+08	**7.67E + 07**
	2.77E+07	1.45E+07	1.49E+07	2.61E+07	2.42E+07	2.41E+07	**4.55E + 06**
	4,-	3,-	2,-	6,-	7,-	5,-	**1**
*F*15	7.11E+03	1.53E+04	1.54E+04	5.89E+03	5.88E+03	5.88E+03	**3.14E + 03**
	1.34E+03	3.92E+02	3.59E+02	9.10E+01	8.81E+01	1.03E+02	**5.42E + 01**
	4,-	5,-	6,-	3,-	2,-	2,-	**1**
*F*16	3.76E+02	1.88E+02	5.08E-02	3.08E-12	4.44E-12	**7.39E-13**	3.79E+00
	4.71E+01	2.16E+02	2.54E-01	3.19E-12	4.22E-13	**5.70E-14**	6.26E-01
	7,-	6,-	4,+	2,+	3,+	**1**,+	5
*F*17	1.59E+05	9.03E+05	6.54E+06	4.50E+04	4.73E+04	4.01E+04	**3.74E + 04**
	1.43E+04	5.28E+04	4.63E+05	3.18E+03	2.77E+03	2.85E+03	**1.36E + 02**
	5,-	6,-	7,-	3,-	4,-	2,-	**1**
*F*18	7.09E+03	2.12E+03	2.47E+03	1.34E+09	3.47E+08	1.11E+10	**1.52E + 03**
	4.77E+03	5.18E+02	1.18E+03	4.94E+08	1.39E+08	2.04E+09	**2.93E + 02**
	4-	2,-	3,-	6,-	5,-	7,-	**1**
*F*19	1.36E+06	1.33E+07	1.59E+07	1.74E+06	1.74E+06	1.74E+06	**1.04E + 06**
	7.35E+04	1.05E+06	1.72E+06	8.46E+04	8.46E+04	9.54E+04	**2.85E + 04**
	2,-	4,-	5,-	3,-	3,-	3,-	**1**
*F*20	2.05E+03	**9.91E + 02**	**9.91E + 02**	9.53E+04	8.42E+03	4.87E+07	1.04E+03
	1.80E+02	**2.61E + 01**	**3.51E + 01**	1.02E+05	2.36E+03	2.27E+07	8.18E+01
	3,-	**1**,+	**1**,+	5,-	4,-	6,-	2
−/+/ =	18/2/0	18/2/0	15/5/0	18/2/0	17/3/0	18/2/0	-
Nb/Nw/Mr	2/6/4.5	1/1/4.45	3/6/3.95	0/2/4.05	1/3/3.9	2/4/3.95	**12/0/1.75**

The proposed WGA algorithm and the other 5 algorithms are applied in these two real-world problems. For comparative studies, FEs_{max} are adjusted to 5.00E+04 and a large enough population size is chosen for all algorithms. Table 9 presents the optimization results (mean and standard deviation) of different algorithms executed in 30 runs for solving the two problems. The best results are shown in boldface, which indicate that WGA provides efficient and better performance compared to the other 5 advanced algorithms for real-world optimization problems.

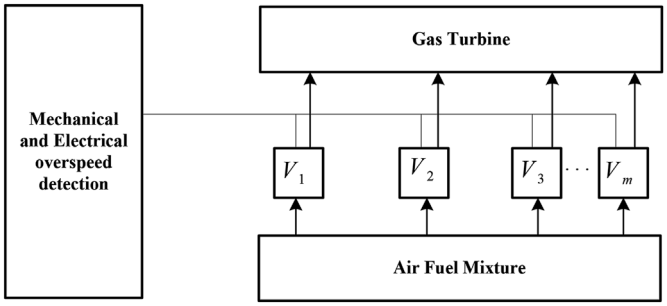

Fig. 4. The diagram block for a gas turbine's overspeed protection system.

Table 9
Average fitness values and standard deviations on real-world optimization problems.

Algorithms	Problem 1		Problem 2	
	Mean	Std	Mean	Std
GL-25	4.05E+000	9.83E+000	8.634E-001	8.114E-001
SaDE	2.72E+000	6.65E+000	8.898E-001	2.875E-002
CoDE	3.19E+000	8.54E+000	8.882E-001	6.155E-001
SPSO2013	7.64E+000	1.15E+001	8.730E-001	6.058E-001
HCLPSO	5.38E+000	1.29E+001	8.875E-001	1.464E-001
WGA	**1.23E-007**	**1.08E-007**	**8.915E-001**	**9.628E-004**

4. Conclusion

The proposed Wild Goose Algorithm (WGA) is a simple and effective algorithm that has been designed and proposed for optimization of high-dimensional problems. This algorithm, which is inspired by wild geese found in nature, includes ordered and coordinated group migration, reproduction and evolution of geese, and also death in the population of geese. To show the performance of the proposed WGA algorithm for optimization of high-dimension problems, it is tested and compared with sep-CMA-ES, CCPSO2, CSO, EPUS-PSO, MLCC, DECCD, DECC-DML, CBCC2-DG, CBCC1-DG and DECC-DG algorithms based on the functions of CEC 2008 and CEC 2010. One of the advantages of WGA is that it has only one control parameter, *Cr*. It is experimentally shown that WGA has better competitive results with respect to other mentioned algorithms, and outperforms all other algorithms for most of the test functions. Furthermore, WGA is a simple and basic algorithm for large-scale optimization which can be used for various real-world optimization problems. In recent years, numerous studies have been carried out in the area of high-dimension optimization, the most of which focused on cooperative co-evolution technique. In future, WGA may be embedded into the frameworks of different CC methods with various categories in order to improve its performance. Furthermore, WGA can be used for solving other real-world large-scale optimization problems.

Credit author statement

Mojtaba Ghasemi: Conceptualization, Methodology, Software, Writing – original draft preparation. Abolfazl Rahimnejad: Data curation, Software, Writing – original draft preparation. Rasul Hemmati: Writing – original draft preparation, Visualization, Investigation. Ebrahim Akbari: Software, Validation. S. Andrew Gadsden: Writing- Reviewing and Editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We would like to acknowledge funding and support from the Natural Sciences and Engineering Research Council (NSERC) Discovery Grant (Gadsden).

References

[1] Mahdavi S, Shiri ME, Rahnamayan S. Metaheuristics in large-scale global continues optimization: a survey. Inf Sci 2015;295:407–28. https://doi.org/10.1016/j.ins.2014.10.042.
[2] Li X, Yao X. Cooperatively coevolving particle swarms for large scale optimization. IEEE Trans Evol Comput 2012;16:210–24. https://doi.org/10.1109/tevc.2011.2112662.
[3] MacNish C, Yao X. Direction matters in high-dimensional optimisation. In: IEEE congr evol comput (IEEE world congr comput intell 2008; 2008. https://doi.org/10.1109/cec.2008.4631115.
[4] Ali MZ, Awad NH, Suganthan PN. Multi-population differential evolution with balanced ensemble of mutation strategies for large-scale global optimization. Appl Soft Comput 2015;33:304–27. https://doi.org/10.1016/j.asoc.2015.04.019.
[5] Tang K, Yáo X, Suganthan PN, MacNish C, Chen Y-P, Chen C-M, et al. Benchmark functions for the CEC'2008 special session and competition on large scale global optimization. Nat Inspired Comput Appl Lab USTC, China 2007;24:1–18.
[6] Tang K, Li X, Suganthan PN, Yang Z, Weise T. Benchmark functions for the CEC 2010 special session and competition on large-scale global optimization. Nat Inspired Comput Appl Lab USTC, China n.d 2010.
[7] Yang Z, Tang K, Yao X. Multilevel cooperative coevolution for large scale optimization. In: IEEE congr evol comput (IEEE world congr comput intell 2008; 2008. https://doi.org/10.1109/cec.2008.4631014.
[8] Hsieh S-T, Sun T-Y, Liu C-C, Tsai S-J. Solving large scale global optimization using improved Particle Swarm Optimizer. IEEE Congr Evol Comput. 2008. https://doi.org/10.1109/cec.2008.4631030. IEEE World Congr Comput Intell 2008.
[9] Ros R, Hansen N. A simple modification in CMA-ES achieving linear time and space complexity. Parallel Probl Solving from Nat – PPSN X 2008:296–305. https://doi.org/10.1007/978-3-540-87700-4_30.
[10] Weber M, Neri F, Tirronen V. Shuffle or update parallel differential evolution for large-scale optimization. Soft Comput 2010;15:2089–107. https://doi.org/10.1007/s00500-010-0640-9.
[11] Chen W, Weise T, Yang Z, Tang K. Large-scale global optimization using cooperative coevolution with variable interaction learning. Parallel probl solving from nature. PPSN XI 2010. https://doi.org/10.1007/978-3-642-15871-1_31. 300–9.
[12] Omidvar MN, Li X, Yao X. Cooperative Co-evolution with delta grouping for large scale non-separable function optimization. In: IEEE congr evol comput; 2010. https://doi.org/10.1109/cec.2010.5585979.
[13] Wang H, Wu Z, Rahnamayan S. Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems. Soft Comput 2010;15:2127–40. https://doi.org/10.1007/s00500-010-0642-7.
[14] Hedar A-R, Ali AF. Tabu search with multi-level neighborhood structures for high dimensional problems. Appl Intell 2011;37:189–206. https://doi.org/10.1007/s10489-011-0321-0.
[15] Chu W, Gao X, Sorooshian S. A new evolutionary search strategy for global optimization of high-dimensional problems. Inf Sci 2011;181:4909–27. https://doi.org/10.1016/j.ins.2011.06.024.
[16] Takahama T, Sakai S. Large scale optimization by differential evolution with landscape modality detection and a diversity archive. In: IEEE congr evol comput; 2012. https://doi.org/10.1109/cec.2012.6252911. 2012.
[17] Wang C, Gao J-H. A differential evolution algorithm with cooperative coevolutionary selection operation for high-dimensional optimization. Opt Lett 2012;8:477–92. https://doi.org/10.1007/s11590-012-0592-3.
[18] Chowdhury JG, Chowdhury A, Sur A. Large scale optimization based on Co-ordinated bacterial dynamics and opposite numbers. Swarm. Evol Memetic Comput 2012;770–7. https://doi.org/10.1007/978-3-642-35380-2_90.
[19] Wang H, Rahnamayan S, Wu Z. Parallel differential evolution with self-adapting control parameters and generalized opposition-based learning for solving high-dimensional optimization problems. J Parallel Distr Comput 2013;73:62–73. https://doi.org/10.1016/j.jpdc.2012.02.019.
[20] Wang Y, Huang J, Dong WS, Yan JC, Tian CH, Li M, et al. Two-stage based ensemble optimization framework for large-scale global optimization. Eur J Oper Res 2013;228:308–20. https://doi.org/10.1016/j.ejor.2012.12.021.
[21] Fan J, Wang J, Han M. Cooperative coevolution for large-scale optimization based on Kernel Fuzzy clustering and variable trust region methods. IEEE Trans Fuzzy Syst 2014;22:829–39. https://doi.org/10.1109/tfuzz.2013.2276863.
[22] Omidvar MN, Li X, Mei Y, Yao X. Cooperative Co-evolution with differential grouping for large scale optimization. IEEE Trans Evol Comput 2014;18:378–93. https://doi.org/10.1109/tevc.2013.2281543.
[23] Segura C, Coello Coello CA, Hernández-Díaz AG. Improving the vector generation strategy of Differential Evolution for large-scale optimization. Inf Sci 2015;323:106–29. https://doi.org/10.1016/j.ins.2015.06.029.
[24] Cheng R, Jin Y. A competitive swarm optimizer for large scale optimization. IEEE Trans Cybern 2015;45:191–204. https://doi.org/10.1109/tcyb.2014.2322602.
[25] Singh D, Agrawal S. Self organizing migrating algorithm with quadratic interpolation for solving large scale global optimization problems. Appl Soft Comput J 2016;38:1040–8. https://doi.org/10.1016/j.asoc.2015.09.033.

[26] Sun Y, Wang X, Chen Y, Liu Z. A modified whale optimization algorithm for large-scale global optimization problems. Expert Syst Appl 2018;114:563–77. https://doi.org/10.1016/j.eswa.2018.08.027.
[27] Li J, Guo L, Li Y, Liu C. Enhancing elephant herding optimization with novel individual updating strategies for large-scale optimization problems. Mathematics 2019;7:395. https://doi.org/10.3390/math7050395.
[28] Shadravan S, Naji HR, Bardsiri VK. The Sailfish Optimizer: a novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. Eng Appl Artif Intell 2019;80:20–34. https://doi.org/10.1016/j.engappai.2019.01.001.
[29] Samareh Moosavi SH, Bardsiri VK. Poor and rich optimization algorithm: a new human-based and multi populations algorithm. Eng Appl Artif Intell 2019;86:165–81. https://doi.org/10.1016/j.engappai.2019.08.025.
[30] Cai X, Zhang J, Liang H, Wang L, Wu Q. An ensemble bat algorithm for large-scale optimization. Int J Mach Learn Cybern 2019;10:3099–113. https://doi.org/10.1007/s13042-019-01002-8.
[31] Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S. Equilibrium optimizer: a novel optimization algorithm. Knowl Base Syst 2020;191:105190. https://doi.org/10.1016/j.knosys.2019.105190.
[32] Salih SQ, Alsewari ARA. A new algorithm for normal and large-scale optimization problems: nomadic People Optimizer. Neural Comput Appl 2020;32:10359–86. https://doi.org/10.1007/s00521-019-04575-1.
[33] Baş E, Ülker E. Improved social spider algorithm for large scale optimization. Artif Intell Rev 2020:1–36. https://doi.org/10.1007/s10462-020-09931-5.
[34] García-Martínez C, Lozano M, Herrera F, Molina D, Sánchez AM. Global and local real-coded genetic algorithms based on parent-centric crossover operators. Eur J Oper Res 2008;185:1088–113. https://doi.org/10.1016/j.ejor.2006.06.043.
[35] Qin AK, Huang VL, Suganthan PN. Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evol Comput 2009;13:398–417. https://doi.org/10.1109/tevc.2008.927706.
[36] Wang Y, Cai Z, Zhang Q. Differential evolution with composite trial vector generation strategies and control parameters. IEEE Trans Evol Comput 2011;15:55–66. https://doi.org/10.1109/tevc.2010.2087271.
[37] Zambrano-Bigiarini M, Clerc M, Rojas R. Standard particle swarm optimisation 2011 at CEC-2013: a baseline for future PSO improvements. In: IEEE congr. Evol. Comput., IEEE; 2013; 2013. p. 2337–44. https://doi.org/10.1109/CEC.2013.6557848.
[38] Lynn N, Suganthan PN. Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation. Swarm Evol Comput 2015;24:11–24. https://doi.org/10.1016/j.swevo.2015.05.002.
[39] Das S, Suganthan PN. Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. Jadavpur Univ Nanyang Technol Univ Kolkata; 2010. p. 341–59.
[40] Chen T-C. IAs based approach for reliability redundancy allocation problems. Appl Math Comput 2006;182:1556–67. https://doi.org/10.1016/j.amc.2006.05.044.
[41] Das S, Abraham A, Chakraborty UK, Konar A. Differential evolution using a neighborhood-based mutation operator. IEEE Trans Evol Comput 2009;13:526–53. https://doi.org/10.1109/tevc.2008.2009457.
[42] Wang H, Rahnamayan S, Sun H, Omran MGH. Gaussian bare-bones differential evolution. IEEE Trans Cybern 2013;43:634–47. https://doi.org/10.1109/tsmcb.2012.2213808.
[43] Zhang H, Hu X, Shao X, Li Z, Wang Y. IPSO-based hybrid approaches for reliability-redundancy allocation problems. Sci China Technol Sci 2013;56:2854–64. https://doi.org/10.1007/s11431-013-5372-5.